

Multi-state Modeling of Biomolecules

Melanie I. Stefan^{1‡*}, Thomas M. Bartol², Terrence J. Sejnowski², Mary B. Kennedy^{3*}

1 Division of Biology and Biological Engineering, California Institute of Technology, Pasadena, California, United States of America, **2** Salk Institute for Biological Studies, La Jolla, California, United States of America, **3** Division of Biology and Biological Engineering, California Institute of Technology, Pasadena, California, United States of America



Abstract: Multi-state modeling of biomolecules refers to a series of techniques used to represent and compute the behavior of biological molecules or complexes that can adopt a large number of possible functional states. Biological signaling systems often rely on complexes of biological macromolecules that can undergo several functionally significant modifications that are mutually compatible. Thus, they can exist in a very large number of functionally different states. Modeling such multi-state systems poses two problems: the problem of how to describe and specify a multi-state system (the “specification problem”) and the problem of how to use a computer to simulate the progress of the system over time (the “computation problem”). To address the specification problem, modelers have in recent years moved away from explicit specification of all possible states and towards rule-based formalisms that allow for implicit model specification, including the κ -calculus [1], BioNetGen [2–5], the Allosteric Network Compiler [6], and others [7,8]. To tackle the computation problem, they have turned to particle-based methods that have in many cases proved more computationally efficient than population-based methods based on ordinary differential equations, partial differential equations, or the Gillespie stochastic simulation algorithm [9,10]. Given current computing technology, particle-based methods are sometimes the only possible option. Particle-based simulators fall into two further categories: nonspatial simulators, such as StochSim [11], DYNSTOC [12], RuleMonkey [9,13], and the Network-Free Stochastic Simulator (NFSim) [14], and spatial simulators, including Meredys [15], SRSim [16,17], and MCell [18–20]. Modelers can thus choose from a variety of tools, the best choice depending on the particular problem. Development of faster and more powerful methods is ongoing, promising the ability to simulate ever more complex signaling processes in the future.

This is a “Topic Page” article for *PLOS Computational Biology*.

Introduction

Multi-state biomolecules in signal transduction

In living cells, signals are processed by networks of proteins that can act as complex computational devices [21]. These networks rely on the ability of single proteins to exist in a variety of functionally different states achieved through multiple mechanisms, including post-translational modifications, ligand binding, conformational change, or formation of new complexes [21–24]. Similarly, nucleic acids can undergo a variety of transformations,

including protein binding, binding of other nucleic acids, conformational change, and DNA methylation.

In addition, several types of modifications can coexist, exerting a combined influence on a biological macromolecule at any given time. Thus, a biomolecule or complex of biomolecules can often adopt a very large number of functionally distinct states. The number of states scales exponentially with the number of possible modifications, a phenomenon known as “combinatorial explosion” [24]. This is of concern for computational biologists who model or simulate such biomolecules, because it raises questions about how such large numbers of states can be represented and simulated.

Examples of combinatorial explosion

Biological signaling networks incorporate a wide array of reversible interactions, post-translational modifications, and conformational changes. Furthermore, it is common for a protein to be composed of several—identical or nonidentical—subunits and for several proteins and/or nucleic acid species to assemble into larger complexes. A molecular species with several of those features can therefore exist in a large number of possible states.

For instance, it has been estimated that the yeast scaffold protein Ste5 can be a part of 25,666 unique protein complexes [22]. In *E. coli*, chemotaxis receptors of four different kinds interact in groups of three, and each individual receptor can exist in at least two possible conformations and has up to eight methylation sites [23], resulting in more than 10^9 potential states. The Ca²⁺/calmodulin-dependent protein kinase II (CaMKII) is a dodecamer of twelve catalytic subunits [25], arranged in two hexameric rings [26]. Each subunit can exist in at least two distinct conformations, and each subunit features various phosphorylation and ligand binding sites. A recent model [27] incorporated conformational states, two

Citation: Stefan MI, Bartol TM, Sejnowski TJ, Kennedy MB (2014) Multi-state Modeling of Biomolecules. *PLoS Comput Biol* 10(9): e1003844. doi:10.1371/journal.pcbi.1003844

Editor: Shoshana Wodak, University of Toronto, Canada

Published: September 25, 2014

Copyright: © 2014 Stefan et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Funding: This work was supported by CRCNS grant DA030749 from NIH (to MBK and TJS) and the Gordon and Betty Moore Foundation “Moore Center for Integrative Study of Cell Regulation” at Caltech. MIS has been supported by a long-term fellowship from EMBO. TMB and TJS acknowledge funding from the Center for Theoretical Biological Physics (NSF PHY-0822283), NIH (P41-GM103712, MH079076, GM086883), and HHMI. The funders had no role in the preparation of this manuscript.

Competing Interests: The authors have declared that no competing interests exist.

* Email: melanie_stefan@hms.harvard.edu (MIS); kennedym@its.caltech.edu (MBK)

‡ **Current address:** Department of Neurobiology, Harvard Medical School, Boston, Massachusetts, United States of America

phosphorylation sites, and two modes of binding calcium/calmodulin, for a total of around 10^9 possible states per hexameric ring. A model of coupling of the EGF receptor to a mitogen-activated protein (MAP) kinase cascade presented by Danos and colleagues [28] accounts for $\sim 10^{23}$ distinct molecular species, yet the authors note several points at which the model could be further extended. A more recent model of ErbB receptor signaling even accounts for more than one googol (10^{100}) distinct molecular species [29]. The problem of combinatorial explosion is also relevant to synthetic biology, with a recent model of a relatively simple synthetic eukaryotic gene circuit featuring 187 species and 1,165 reactions [30].

Of course, not all of the possible states of a multi-state molecule or complex will necessarily be populated. Indeed, in systems in which the number of possible states is far greater than that of molecules in the compartment (e.g., the cell), they cannot be. In some cases, empirical information can be used to rule out certain states if, for instance, some combinations of features are incompatible. In the absence of such information, however, all possible states need to be considered a priori. In such cases, computational modeling can be used to uncover to what extent the different states are populated.

It is worth noting that the existence (or potential existence) of such large numbers of molecular species is a combinatorial phenomenon: it arises from a relatively small set of features or modifications (such as post-translational modification or complex formation) that combine to dictate the state of the entire molecule or complex in the same way that the existence of just a few choices in a coffee shop (small, medium, or large; with or without milk; decaf or not; extra shot of espresso) quickly leads to a large number of possible beverages (24 in this case; each additional binary choice will double that number). Although it is difficult for us to grasp the total number of possible combinations, it is usually not conceptually difficult to understand the (much smaller) set of features or modifications and the effect each of them has on the function of the biomolecule. The rate at which a molecule undergoes a particular reaction will usually depend mainly on a single feature or a small subset of features. It is the presence or absence of those features that dictates the reaction rate. The reaction rate is the same for two molecules that differ only in features that do not affect this reaction. Thus, the number of parameters will be much smaller than the number of reactions. (In the coffee shop example, adding an extra shot of espresso will cost 40 cents, no matter what size the beverage is and whether or not it has milk in it). It is such “local rules” that are usually discovered in laboratory experiments. Thus, a multi-state model can be conceptualized in terms of combinations of modular features and local rules. This means that even a model that can account for a vast number of molecular species and reactions is not necessarily conceptually complex.

Specification versus computation

The combinatorial complexity of signaling systems involving multi-state proteins poses two kinds of problems. The first problem is concerned with how such a system can be specified, i.e., how a modeler can specify all complexes, all changes those complexes undergo, and all parameters and conditions governing those changes in a robust and efficient way. This problem is called the “specification problem.” The second problem concerns computation. It asks questions about whether a combinatorially complex model, once specified, is computationally tractable given the large number of states and the even larger number of possible transitions between states, whether it can be stored electronically, and whether it can be evaluated in a reasonable amount of computing time. This problem is called the “computation

problem.” Among the approaches that have been proposed to tackle combinatorial complexity in multi-state modeling, some are mainly concerned with addressing the specification problem, and some are focused on finding effective methods of computation. Some tools address both specification and computation. The sections below discuss rule-based approaches to the specification problem and particle-based approaches to solving the computation problem. A list of the tools discussed here is presented in Figure 1. A comprehensive overview and discussion of various tools available for multi-state modeling can be found in Chylek et al. [31].

The Specification Problem

Explicit specification

The most naïve way of specifying a biomolecule in a biological model is to specify each of its states explicitly and use each of them as a molecular species in a simulation framework that allows transitions from state to state. For instance, if a protein can be ligand or not, exist in two conformational states (e.g., open or closed), and be located in two possible subcellular areas (e.g., cytosolic or membrane), then the eight possible resulting states can be explicitly enumerated as follows:

- bound, open, cytosol
- bound, open, membrane
- bound, closed, cytosol
- bound, closed, membrane
- unbound, open, cytosol
- unbound, open, membrane
- unbound, closed, cytosol
- unbound, closed, membrane

Enumerating all possible states is a lengthy and potentially error-prone process. For macromolecular complexes that can adopt multiple states, enumerating each state quickly becomes tedious, if not impossible. Moreover, the addition of a single additional modification or feature to the model of the complex under investigation will double the number of possible states (if the modification is binary), and it will more than double the number of transitions that need to be specified.

Rule-based model specification

It is clear that an explicit description, which lists all possible molecular species (including all their possible states), all possible reactions or transitions these species can undergo, and all parameters governing these reactions, very quickly becomes unwieldy as the complexity of the biological system increases. Modelers have therefore looked for implicit, rather than explicit, ways of specifying a biological signaling system. An implicit description is one that groups reactions and parameters that apply to many types of molecular species into one reaction template. It might also add a set of conditions that govern reaction parameters, e.g., the likelihood or rate at which a reaction occurs or whether it occurs at all. Only properties of the molecule or complex that matter to a given reaction (either affecting the reaction or being affected by it) are explicitly mentioned, and all other properties are ignored in the specification of the reaction.

For instance, the rate of ligand dissociation from a protein might depend on the conformational state of the protein but not on its subcellular localization. An implicit description would therefore list two dissociation processes (with different rates, depending on conformational state) but would ignore attributes referring to

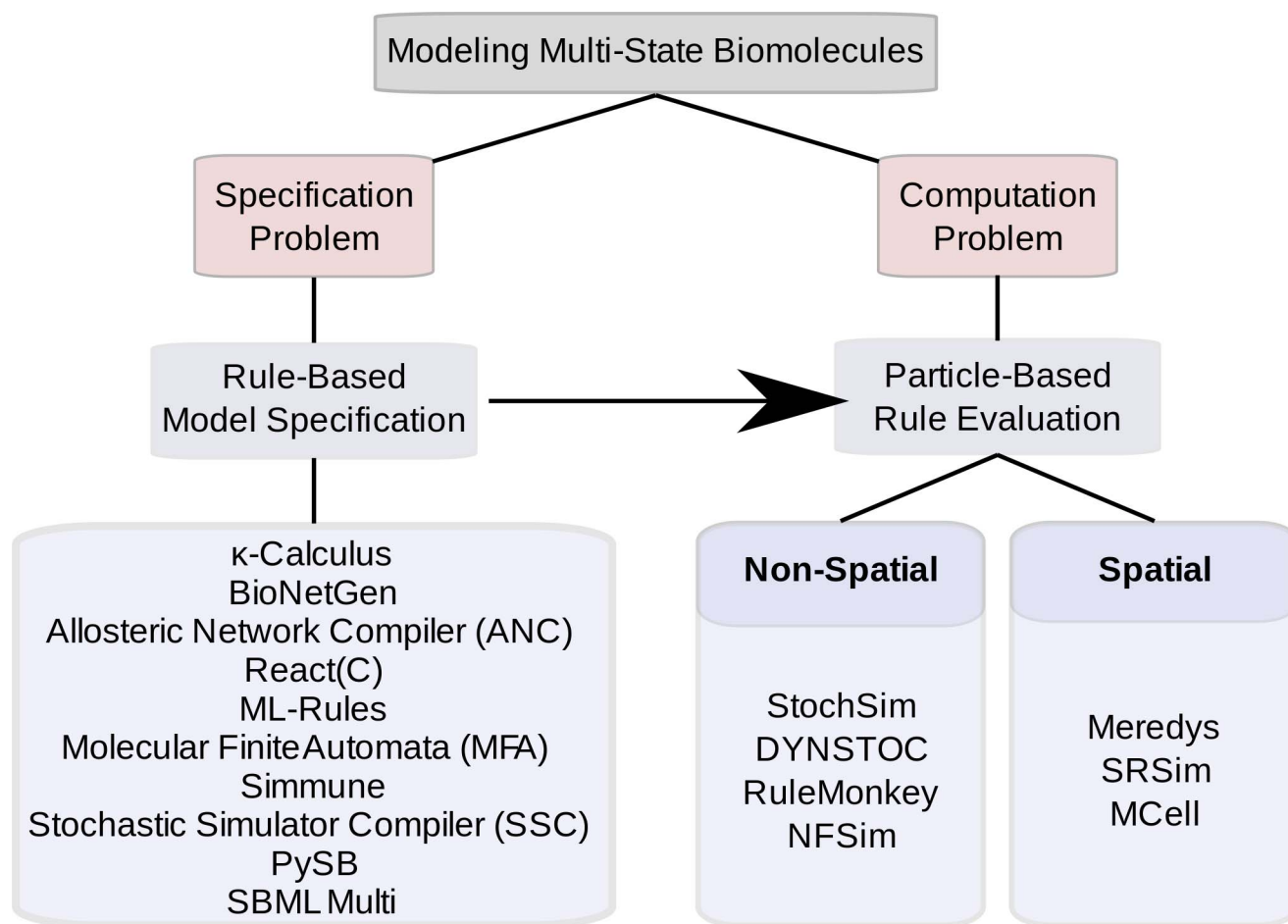


Figure 1. An overview of tools discussed here that are used for the rule-based specification and particle-based evaluation (spatial or nonspatial) of multi-state biomolecules.

doi:10.1371/journal.pcbi.1003844.g001

subcellular localization, because they do not affect the rate of ligand dissociation nor are they affected by it. This specification rule has been summarized as “don’t care, don’t write” [28].

Since it is not written in terms of reactions but in terms of more general “reaction rules” encompassing sets of reactions, this kind of specification is often called “rule-based” [4]. This description of the system in terms of modular rules relies on the assumption that only a subset of features or attributes are relevant for a particular reaction rule. Where this assumption holds, a set of reactions can be coarse-grained into one reaction rule. This coarse-graining preserves the important properties of the underlying reactions. For instance, if the reactions are based on chemical kinetics, so are the rules derived from them.

Many rule-based specification methods exist. In general, the specification of a model is a separate task from the execution of the simulation. Therefore, among the existing rule-based model specification systems [4], some concentrate on model specification only, allowing the user to then export the specified model into a dedicated simulation engine. However, many solutions to the specification problem also contain a method of interpreting the specified model [3]. This is done by providing a method to simulate the model or a method to convert it into a form that can be used for simulations in other programs.

An early rule-based specification method is the Kappa (κ)-calculus [1], a process algebra that can be used to encode

macromolecules with internal states and binding sites and to specify rules by which they interact. A review of κ is provided by Danos et al. [28]. The κ -calculus is merely concerned with providing a language to encode multi-state models, not with interpreting the models themselves. A simulator compatible with Kappa is KaSim [32,33].

BioNetGen is a software suite that provides both specification and simulation capacities [2–5]. Rule-based models can be written down using a specified syntax, the BioNetGen language (BNGL) [4]. The underlying concept is to represent biochemical systems as graphs, in which molecules are represented as nodes (or collections of nodes) and chemical bonds as edges. A reaction rule then corresponds to a graph rewriting rule [3]. BNGL provides a syntax for specifying these graphs and the associated rules as structured strings [4]. BioNetGen can then use these rules to generate ordinary differential equations (ODEs) to describe each biochemical reaction. Alternatively, it can generate a list of all possible species and reactions in the Systems Biology Markup Language (SBML) [34,35], which can then be exported to simulation software packages that can read SBML. One can also make use of BioNetGen’s own ODE-based simulation software and its capability to generate reactions on the fly during a stochastic simulation [5]. In addition, a model specified in BNGL can be read by other simulation software, such as DYNSTOC [12], RuleMonkey [13], and NFSim [14].

Another tool that generates full reaction networks from a set of rules is the Allosteric Network Compiler (ANC) [6]. Conceptually, ANC sees molecules as allosteric devices with a Monod-Wyman-Changeux (MWC)-type regulation mechanism [36], whose inter-actions are governed by their internal state, as well as by external modifications. A very useful feature of ANC is that it automatically computes dependent parameters, thereby imposing thermodynamic correctness [37].

An extension of the κ -calculus is provided by React(C) [38]. The authors of React(C) show that it can express the stochastic π calculus [39]. They also provide a stochastic simulation algorithm based on the Gillespie stochastic algorithm [40] for models specified in React(C) [38].

ML-Rules [41] is similar to React(C) but provides the added possibility of nesting: a component species of the model, with all its attributes, can be part of a higher-order component species. This enables ML-Rules to capture multi-level models that can bridge the gap between, for instance, a series of biochemical processes and the macroscopic behavior of a whole cell or group of cells. For instance, Maus et al. have provided a proof-of-concept model of cell division in fission yeast that includes cyclin/cdc2 binding and activation, pheromone secretion and diffusion, cell division, and movement of cells [41]. Models specified in ML-Rules can be simulated using the Java Framework for Modeling and Simulation (JAMES) II [42]. A similar nested language to represent multi-level biological systems has been proposed by Oury and Plotkin [43].

Yang et al. [8] have proposed a specification formalism based on finite automata. Models specified in their Molecular Finite Automata (MFA) framework can then be used to generate and simulate a system of ODEs or for stochastic simulation using a kinetic Monte Carlo algorithm.

Some rule-based specification systems and their associated network generation and simulation tools have been designed to accommodate spatial heterogeneity in order to allow for the realistic simulation of interactions within biological compartments. For instance, the Simmune project [44,45] includes a spatial component: users can specify their multi-state biomolecules and interactions within membranes or compartments of arbitrary shape. The reaction volume is then divided into interfacing voxels, and a separate reaction network is generated for each of these subvolumes.

The Stochastic Simulator Compiler (SSC) [46] allows for rule-based, modular specification of interacting biomolecules in regions of arbitrarily complex geometries. Again, the system is represented using graphs, with chemical interactions or diffusion events formalized as graph-rewriting rules [46]. The compiler then generates the entire reaction network before launching a stochastic reaction-diffusion algorithm.

A different approach is taken by PySB [47], in which model specification is embedded in the programming language Python. A model (or part of a model) is represented as a Python program. This allows users to store higher-order biochemical processes such as catalysis or polymerization as macros and reuse them as needed. The models can be simulated and analyzed using Python libraries, but PySB models can also be exported into BNGL [4], Kappa [1], and SBML [34].

Models involving multi-state and multi-component species can also be specified in level 3 of the SBML [34] using the multi package. A draft specification is available [48], and software support is under development.

Thus, by only considering states and features important for a particular reaction, rule-based model specification eliminates the need to explicitly enumerate every possible molecular state that

can undergo a similar reaction and thereby allows for efficient specification.

The Computation Problem

When running simulations on a biological model, any simulation software evaluates a set of rules, starting from a specified set of initial conditions and usually iterating through a series of time steps until a specified end time. One way to classify simulation algorithms is by looking at the level of analysis at which the rules are applied: they can be population-based, single-particle-based, or hybrid.

Population-based rule evaluation

In population-based rule evaluation, rules are applied to populations. All molecules of the same species in the same state are pooled together. Application of a specific rule reduces or increases the size of one of the pools, possibly at the expense of another.

Some of the best-known classes of simulation approaches in computational biology belong to the population-based family, including those based on the numerical integration of ordinary and partial differential equations and the Gillespie stochastic simulation algorithm.

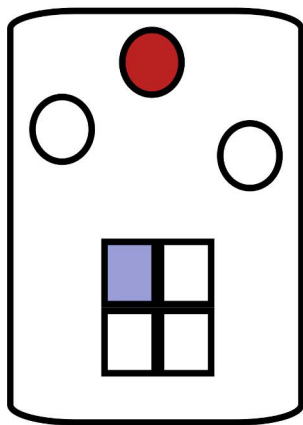
Differential equations describe changes in molecular concentrations over time in a deterministic manner. Simulations based on differential equations usually do not attempt to solve those equations analytically but employ a suitable numerical solver.

The stochastic Gillespie algorithm changes the composition of pools of molecules through a progression of random reaction events, the probability of which is computed from reaction rates and from the numbers of molecules, in accordance with the stochastic master equation [40].

In population-based approaches, one can think of the system being modeled as being in a given state at any given time point, where a state is defined according to the nature and size of the populated pools of molecules. This means that the space of all possible states can become very large. With some simulation methods implementing numerical integration of ordinary and partial differential equations or the Gillespie stochastic algorithm, all possible pools of molecules and the reactions they undergo are defined at the start of the simulation, even if they are empty. Such “generate-first” methods [4] scale poorly with increasing numbers of molecular states [49]. For instance, it has recently been estimated that even for a simple model of CaMKII with just six states per subunits and ten subunits, it would take 290 years to generate the entire reaction network on a 2.54 GHz Intel Xeon processor [50]. In addition, the model generation step in generate-first methods does not necessarily terminate, for instance, when the model includes assembly of proteins into complexes of arbitrarily large size, such as actin filaments. In these cases, a termination condition needs to be specified by the user [3,5].

Even if a large reaction system can be successfully generated, its simulation using population-based rule evaluation can run into computational limits. In a recent study, a powerful computer was shown to be unable to simulate a protein with more than eight phosphorylation sites ($2^8 = 256$ phosphorylation states) using ordinary differential equations [14].

Methods have been proposed to reduce the size of the state space. One is to consider only the states adjacent to the present state (i.e., the states that can be reached within the next iteration) at each time point. This eliminates the need for enumerating all possible states at the beginning. Instead, reactions are generated “on the fly” [4] at each iteration. These methods are available



A is phosphorylated

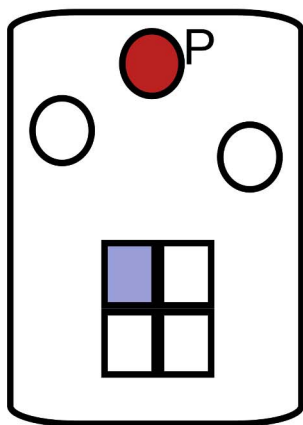
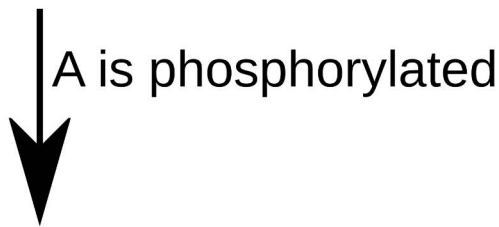
Phospho-A is dephosphorylated

Phospho-A binds to B

A dissociates from B

B bound to phospho-A is activated

B is activated by neighboring subunit



A is phosphorylated

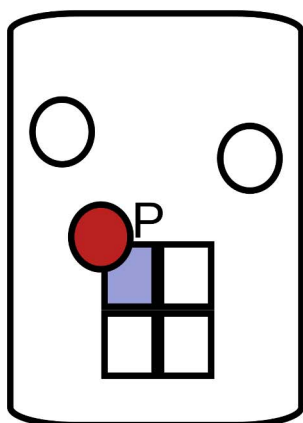
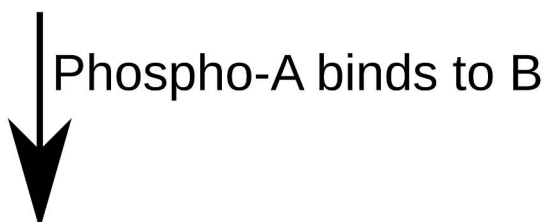
Phospho-A is dephosphorylated

Phospho-A binds to B

A dissociates from B

B bound to phospho-A is activated

B is activated by neighboring subunit



A is phosphorylated

Phospho-A is dephosphorylated

Phospho-A binds to B

A dissociates from B

B bound to phospho-A is activated

B is activated by neighboring subunit

Figure 2. Principles of particle-based modeling. In particle-based modeling, each particle is tracked individually through the simulation. At any point, a particle only “sees” the rules that apply to it. This figure follows two molecular particles (one of type A in red, one of type B in blue) through three steps in a hypothetical simulation following a simple set of rules (given on the right). At each step, the rules that potentially apply to the particle under consideration are highlighted in that particle’s colour.

doi:10.1371/journal.pcbi.1003844.g002

both for stochastic and deterministic algorithms. These methods still rely on the definition of an (albeit reduced) reaction network—in contrast to the “network-free” methods discussed below.

Even with “on-the-fly” network generation, networks generated for population-based rule evaluation can become quite large and thus difficult—if not impossible—to handle computationally. An alternative approach is provided by particle-based rule evaluation.

Particle-based rule evaluation

In particle-based (sometimes called “agent-based”) simulations, proteins, nucleic acids, macromolecular complexes, or small molecules are represented as individual software objects, and their progress is tracked through the course of the entire simulation [51]. Because particle-based rule evaluation keeps track of individual particles rather than populations, it comes at a higher

computational cost when modeling systems with a high total number of particles but a small number of kinds (or pools) of particles [51]. In cases of combinatorial complexity, however, the modeling of individual particles is an advantage because, at any given point in the simulation, only existing molecules, their states, and the reactions they can undergo need to be considered. Particle-based rule evaluation does not require the generation of complete or partial reaction networks at the start of the simulation or at any other point in the simulation and is therefore called “network-free.”

This method reduces the complexity of the model at the simulation stage and thereby saves time and computational power [9]. A detailed discussion of the computational cost of population-based versus particle-based methods is provided in a recent study by Hogg et al. [10]. The simulation follows each particle, and at

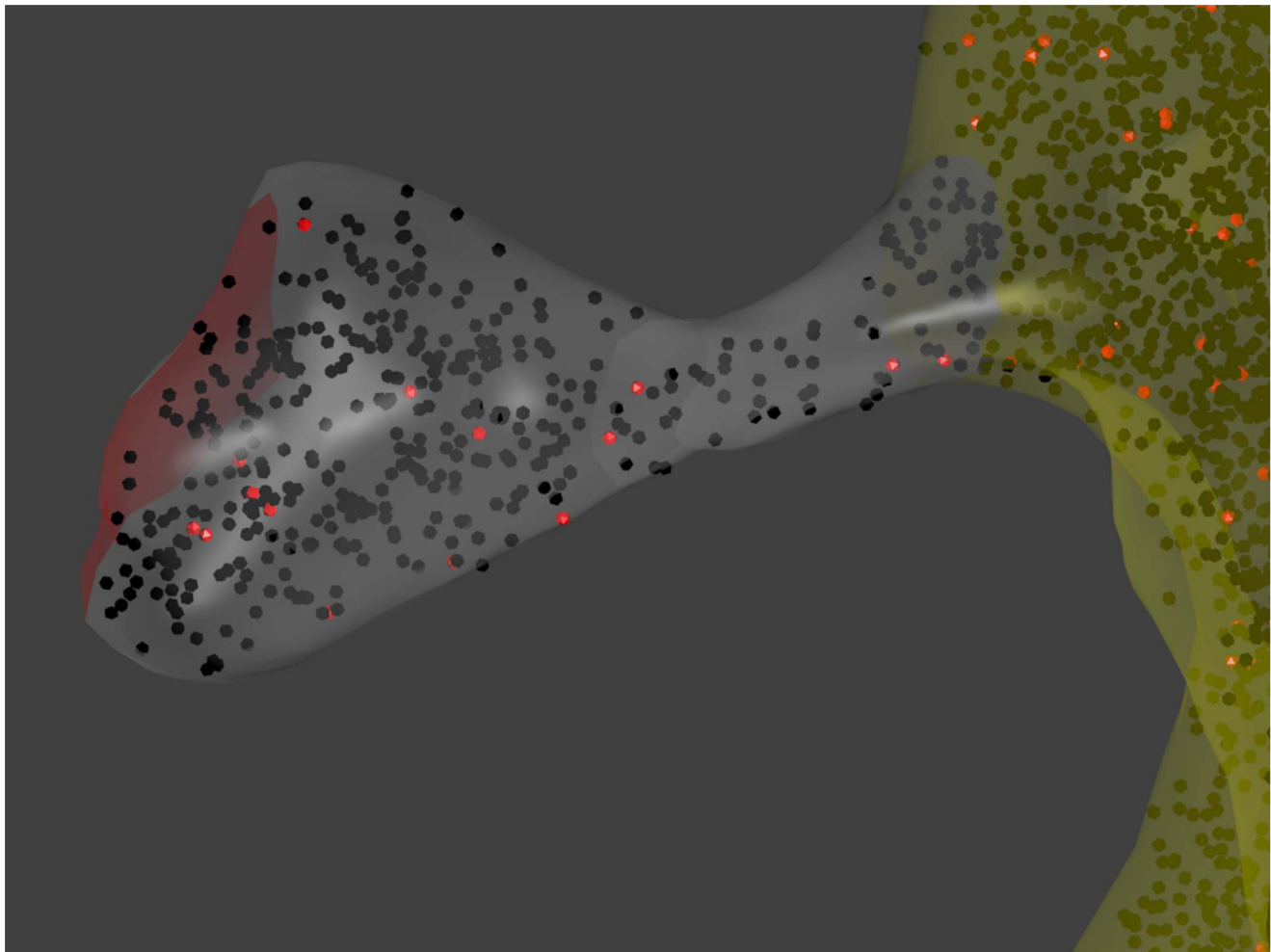


Figure 3. Screenshot from an MCell simulation of calcium signaling within the spine. Although other types of calcium-regulated molecules were included in the simulations, only CaMKII molecules are visualized. They are shown in red when bound to calmodulin and in black when unbound. The simulation compartment is a reconstruction of a dendritic spine as presented by Kinney et al. [59]. The area of the postsynaptic density is shown in red, the spine head and neck in gray, and the parent dendrite in yellow. The figure was generated by visualizing the simulation results in Blender.

doi:10.1371/journal.pcbi.1003844.g003

Table 1. Examples of multi-state models of biological systems.

Biological system	Specification	Computation	Reference
Bacterial chemotaxis signaling pathway	StochSim	StochSim	[61]
CaMKII regulation	StochSim	StochSim	[27]
ERBB receptor signaling	BioNetGen	NFSim	[29]
Eukaryotic synthetic gene circuits	BioNetGen, PROMOT [62]	COPASI [63]	[30]
RNA signaling	Kappa	KaSim	[64]
Cooperativity of allosteric proteins	ANC	Matlab	[6]
Chemosensing in Dictyostelium	Simmune	Simmune	[44]
T cell receptor activation	SSC	SSC	[65]
Human mitotic kinetochore	BioNetGen	SRSim	[66]
Cell cycle of fission yeast	ML-Rules	JAMES II [42]	[41]

A version of this table with hyperlinks is attached to this manuscript as Table S1. Abbreviations: COPASI, COMplex PATHway Simulator; PROMOT, Process Modeling Tool. doi:10.1371/journal.pcbi.1003844.t001

each simulation step, a particle only “sees” the reactions (or rules) that apply to it. This depends on the state of the particle and, in some implementations, on the states of its neighbors in a holoenzyme or complex. As the simulation proceeds, the states of particles are updated according to the rules that are fired. Figure 2 illustrates the process of particle-based modeling using a simple system with three molecules of type A and one molecular tetramer of type B. This system goes through three simulation steps following a simple set of rules.

Some particle-based simulation packages use an ad hoc formalism for specification of reactants, parameters, and rules. Others can read files in a recognized rule-based specification format such as BNGL [4].

Nonspatial particle-based methods

StochSim [11,52] is a particle-based stochastic simulator used mainly to model chemical reactions and other molecular transitions. The algorithm used in StochSim is different from the more widely known Gillespie stochastic algorithm [40] in that it operates on individual entities, not entity pools, making it particle-based rather than population-based.

In StochSim, each molecular species can be equipped with a number of binary state flags representing a particular modification. Reactions can be made dependent on a set of state flags set to particular values. In addition, the outcome of a reaction can include a state flag being changed. Moreover, entities can be arranged in geometric arrays (for instance, for holoenzymes consisting of several subunits), and reactions can be “neighbor-sensitive”, i.e., the probability of a reaction for a given entity is affected by the value of a state flag on a neighboring entity. These properties make StochSim ideally suited to modeling multi-state molecules arranged in holoenzymes or complexes of specified size. Indeed, StochSim has been used to model clusters of bacterial chemotactic receptors [53] and CaMKII holoenzymes [27].

An extension to StochSim has been presented by Colvin et al. [12]. Their particle-based simulator DYNSTOC uses a StochSim-like algorithm to simulate models specified in BNGL [4], which improves the handling of molecules within macromolecular complexes [12].

Another particle-based stochastic simulator that can read BNGL input files is RuleMonkey [13]. Its simulation algorithm [9] differs from the algorithms underlying both StochSim and DYNSTOC in that the simulation time step is variable.

NFSim differs from those described above by allowing for the definition of reaction rates as arbitrary mathematical or condi-

tional expressions and thereby facilitates selective coarse-graining of models [14]. RuleMonkey and NFSim implement distinct but related simulation algorithms. A detailed review and comparison of both tools is given by Yang and Hlavacek [54].

It is easy to imagine a biological system in which some components are complex multi-state molecules, whereas others have few possible states (or even just one) and exist in large numbers. A hybrid approach has been proposed to model such systems: within the hybrid particle/population (HPP) framework, the user can specify a rule-based model but can designate some species to be treated as populations (rather than particles) in the subsequent simulation [10]. This method combines the computational advantages of particle-based modeling for multi-state systems with relatively low molecule numbers and of population-based modeling for systems with high molecule numbers and a small number of possible states. Specification of HPP models is supported by BioNetGen [4], and simulations can be performed with NFSim [14].

Spatial particle-based methods

Spatial particle-based methods differ from the methods described above by their explicit representation of space.

One example of a particle-based simulator that allows for a representation of cellular compartments is SRSim [16,17]. SRSim is integrated in the Large-scale Atomic/Molecular Massively Parallel Simulator (LAMMPS) [55,56] and allows the user to specify the model in BNGL [4]. SRSim allows users to specify the geometry of the particles in the simulation, as well as interaction sites. It is therefore especially good at simulating the assembly and structure of complex biomolecular complexes, as evidenced by a recent model of the inner kinetochore [57].

MCell [18–20,58] allows individual molecules to be traced in arbitrarily complex geometric environments that are defined by the user. This allows for simulations of biomolecules in realistic reconstructions of living cells, including cells with complex geometries like those of neurons. As an illustration, Figure 3 shows a screenshot from a simulation of calcium proteins. The reaction compartment is a reconstruction of a dendritic spine [59]. Visualizations are supported by a specialized plug-in (“CellBlender”) for the open-source program Blender [60].

MCell uses an ad hoc formalism within MCell itself to specify a multi-state model: in MCell, it is possible to assign “slots” to any molecular species. Each slot stands for a particular modification, and any number of slots can be assigned to a molecule. Each slot

can be occupied by a particular state. The states are not necessarily binary. For instance, a slot describing binding of a particular ligand to a protein of interest could take the states “unbound,” “partially bound,” and “fully bound.”

The slot-and-state syntax in MCell can also be used to model multimeric proteins or macromolecular complexes. When used in this way, a slot is a placeholder for a subunit or a molecular component of a complex, and the state of the slot will indicate whether a specific protein component is absent or present in the complex. A way to think about this is that MCell macromolecules can have several dimensions: a “state dimension” and one or more “spatial dimensions.” The “state dimension” is used to describe the multiple possible states making up a multi-state protein, while the spatial dimension(s) describes topological relationships between neighboring subunits or members of a macromolecular complex. One drawback of this method for representing protein complexes, compared to other spatial modeling tools such as Meredys [15], is that MCell does not allow for the diffusion of complexes and hence of multi-state molecules. This can in some cases be circumvented by adjusting the diffusion constants of ligands that interact with the complex by using checkpointing functions or by combining simulations at different levels.

References

- Danos V, Laneve C (2004) Formal molecular biology. *Theor Comput Sci* 325: 69–110.
- Blinov ML, Faeder JR, Goldstein B, Hlavacek WS (2004) BioNetGen: software for rule-based modeling of signal transduction based on the interactions of molecular domains. *Bioinformatics* 20: 3289–3291.
- Faeder JR, Blinov ML, Goldstein B, Hlavacek WS (2005) Rule-Based Modeling of Biochemical Networks. *Complexity* 10: 22–41.
- Hlavacek WS, Faeder JR, Blinov ML, Posner RG, Hucka M, et al. (2006) Rules for modeling signal-transduction systems. *Sci STKE* 2006: re6.
- Faeder JR, Blinov ML, Hlavacek WS (2009) Rule-based modeling of biochemical systems with BioNetGen. *Methods Mol Biol* 500: 113–167.
- Ollivier JF, Shahrezaei V, Swain PS (2010) Scalable rule-based modelling of allosteric proteins and biochemical networks. *PLoS Comput Biol* 6: e1000975.
- Lok L, Brent R (2005) Automatic generation of cellular reaction networks with Molecuizer 1.0. *Nat Biotechnol* 23: 131–136.
- Yang J, Meng X, Hlavacek WS (2010) Rule-based modelling and simulation of biochemical systems with molecular finite automata. *IET Syst Biol* 4: 453–466.
- Yang J, Monine MI, Faeder JR, Hlavacek WS (2008) Kinetic Monte Carlo method for rule-based modeling of biochemical networks. *Phys Rev E Stat Nonlin Soft Matter Phys* 78: 031910.
- Hogg JS, Harris LA, Stover LJ, Nair NS, Faeder JR (2014) Exact Hybrid Particle/Population Simulation of Rule-Based Models of Biochemical Systems. *PLoS Comput Biol* 10: e1003544. doi:10.1371/journal.pcbi.1003544
- Le Novère N, Shimizu TS (2001) STOCHSIM: modelling of stochastic biomolecular processes. *Bioinformatics* 17: 575–576.
- Colvin J, Monine MI, Faeder JR, Hlavacek WS, Von Hoff DD, et al. (2009) Simulation of large-scale rule-based models. *Bioinformatics* 25: 910–917.
- Colvin J, Monine MI, Gutenkunst RN, Hlavacek WS, Von Hoff DD, et al. (2010) RuleMonkey: software for stochastic simulation of rule-based models. *BMC Bioinformatics* 11: 404.
- Sneddon MW, Faeder JR, Emonet T (2011) Efficient modeling, simulation and coarse-graining of biological complexity with NFSim. *Nat Methods* 8: 177–183.
- Tolle DP, Le Novère N (2010) Meredys, a multi-compartment reaction-diffusion simulator using multi-state realistic molecular complexes. *BMC Syst Biol* 4: 24.
- Gruenert G, Ibrahim B, Lenser T, Lohel M, Hinze T, et al. (2010) Rule-based spatial modeling with diffusing, geometrically constrained molecules. *BMC Bioinformatics* 11: 307.
- Grünert G, Dittrich P (2011) Using the SRSim Software for Spatial and Rule-Based Modeling of Combinatorially Complex Biochemical Reaction Systems. *Membrane Computing - Lecture Notes in Computer Science* 6501: 240–256.
- Stiles JR, Van Helden D, Bartol TM, Salpeter EE, Salpeter MM (1996) Miniature endplate current rise times less than 100 microseconds from improved dual recordings can be modeled with passive acetylcholine diffusion from a synaptic vesicle. *Proc Natl Acad Sci U S A* 93: 5747–5752.
- Stiles JR, Bartol TM (2001) Monte Carlo methods for simulating realistic synaptic microphysiology using MCell. In: De Schutter E, editor. *Computational Neuroscience: Realistic Modeling for Experimentalists*. Boca Raton (Florida): CRC Press.
- Kerr RA, Bartol TM, Kaminsky B, Dittrich M, Chang JJ, et al. (2008) Fast Monte Carlo simulation methods for biological reaction-diffusion systems in solution and on surfaces. *SIAM J Sci Comput* 30: 3126.

Examples of Multi-state Models in Biology

A (by no means exhaustive) selection of models of biological systems involving multi-state molecules and using some of the tools discussed here is given in Table 1.

Supporting Information

Table S1 Table 1 with hyperlinks.
(DOCX)

Text S1 Version history of the text file.
(XML)

Text S2 Peer reviews and response to reviews. Human-readable versions of the reviews and authors responses are available as comments on this article.
(XML)

Acknowledgments

The version history of the text file and the peer reviews (and response to reviews) are available as supporting information in Text S1 and S2.

- Bray D (1995) Protein molecules as computational elements in living cells. *Nature* 376: 307–312.
- Endy D, Brent R (2001) Modelling cellular behaviour. *Nature* 409: 391–395.
- Bray D (2003) Genomics. Molecular prodigality. *Science* 299: 1189–1190.
- Hlavacek WS, Faeder JR, Blinov ML, Perelson AS, Goldstein B (2003) The complexity of complexes in signal transduction. *Biotechnol Bioeng* 84: 783–794.
- Bennett MK, Erondy NE, Kennedy MB (1983) Purification and characterization of a calmodulin-dependent protein kinase that is highly concentrated in brain. *J Biol Chem* 258: 12735–12744.
- Rosenberg OS, Deindl S, Sung R, Nairn AC, Kuriyan J (2005) Structure of the autoinhibited kinase domain of CaMKII and SAXS analysis of the holoenzyme. *Cell* 123: 849–860.
- Stefan MI, Marshall DP, Le Novère N (2012) Structural analysis and stochastic modelling suggest a mechanism for calmodulin trapping by CaMKII. *PLoS ONE* 7: e29406.
- Danos V, Feret J, Fontana W, Harmer R, Krivine J (2007) Rule-Based Modelling of Cellular Signalling. In: *Proceedings of the Eighteenth International Conference on Concurrency Theory, CONCUR 2007; September 3–8, 2007; Lisbon, Portugal*.
- Creamer MS, Stites EC, Aziz M, Cahill JA, Tan CW, et al. (2012) Specification, annotation, visualization and simulation of a large rule-based model for ERBB receptor signaling. *BMC Syst Biol* 6: 107.
- Marchisio MA, Colaiacovo M, Whitehead E, Stelling J (2013) Modular, rule-based modeling for the design of eukaryotic synthetic gene circuits. *BMC Syst Biol* 7: 42.
- Chylek LA, Stites EC, Posner RG, Hlavacek WS (2013) Innovations of the rule-based modeling approach. In: Prokop A, Csukás B, editors. *Systems Biology: Integrative Biology and Simulation Tools Volume 1*. Dordrecht: Springer. 553 p.
- Feret J, Danos V, Krivine J, Harmer R, Fontana W (2009) Internal coarse-graining of molecular systems. *Proc Natl Acad Sci U S A* 106: 6453–6458.
- Krivine J (2014) KaSim. Available: <https://github.com/jkrivine/KaSim>. Accessed 22 August 2014.
- Hucka M, Finney A, Sauro HM, Bolouri H, Doyle JC, et al. (2003) The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. *Bioinformatics* 19: 524–531.
- Finney A, Hucka M (2003) Systems biology markup language: Level 2 and beyond. *Biochem Soc Trans* 31: 1472–1473.
- Monod J, Wyman J, Changeux JP (1965) On the nature of allosteric transitions: a plausible model. *J Mol Biol* 12: 88–118.
- Colquhoun D, Dowdsland KA, Beato M, Plested AJR (2004) How to impose microscopic reversibility in complex reaction mechanisms. *Biophys J* 86: 3510–3518.
- John M, Lhoussaine C, Niehren J, Versari C (2011) Biochemical reaction rules with constraints. In: *Programming Languages and Systems*. Berlin Heidelberg: Springer. pp. 338–357.
- Priami C (1995) Stochastic π -calculus. *The Computer Journal* 38: 578–589.
- Gillespie DT (1977) Exact Stochastic Simulation of Coupled Chemical Reactions. *J Phys Chem* 81: 2340–2361.
- Maus C, Rybacki S, Uhrmacher AM (2011) Rule-based multi-level modeling of cell biological systems. *BMC Syst Biol* 5: 166.

42. Himmelspach J, Uhrmacher AM (2007) "Plug'n simulate." In: Proceedings of the 40th Annual Simulation Symposium; March 2007; Norfolk, Virginia. IEEE Computer Society. pp. 137–143.
43. Oury N, Plotkin G (2013) Multi-level modelling via stochastic multi-level multiset rewriting. *Math Structures Comput Sci* 23: 471–503.
44. Meier-Schellersheim M, Xu X, Angermann B, Kunkel EJ, Jin T, et al. (2006) Key role of local regulation in chemosensing revealed by a new molecular interaction-based modeling method. *PLoS Comput Biol* 2: e82.
45. Angermann BR, Klauschen F, Garcia AD, Prustel T, Zhang F, et al. (2012) Computational modeling of cellular signaling processes embedded into dynamic spatial contexts. *Nat Methods* 9: 283–289.
46. Mieszko L, Artyomov MN, Devadas S, Chakraborty AK (2009) Efficient stochastic simulation of reaction-diffusion processes via direct compilation. *Bioinformatics* 25: 2289–2291.
47. Lopez CF, Muhlich JL, Bachman JA, Sorger PK (2013) Programming biological models in Python using PySB. *Mol Syst Biol* 9: 646.
48. Zhang F, Meier-Schellersheim M (2013) SBML Level 3 Package Specification: Multi-state, Multicomponent and Multicompartment Species Package for SBML Level 3 (Multi). Version 1, Release 01 (Draft, Rev 369). Available: http://sbml.org/Documents/Specifications/SBML_Level_3/Packages/multi. Accessed 22 August 2014.
49. Tolle DP, Le Novère N (2006) Particle-Based Stochastic Simulation in Systems Biology. *Curr Bioinformatics* 1: 315–320.
50. Michalski PJ, Loew LM (2012) CaMKII activation and dynamics are independent of the holoenzyme structure: an infinite subunit holoenzyme approximation. *Phys Biol* 9: 036010.
51. Mogilner A, Allard J, Wollman R (2012) Cell polarity: quantitative modeling as a tool in cell biology. *Science* 336: 175–179.
52. Le Novère N (2013) Stochsim. Available: <http://sourceforge.net/projects/stochsim/>. Accessed 22 August 2014.
53. Levin MD, Shimizu TS, Bray D (2002) Binding and diffusion of CheR molecules within a cluster of membrane receptors. *Biophys J* 82: 1809–1817.
54. Yang J, Hlavacek WS (2011) The efficiency of reactant site sampling in network-free simulation of rule-based models for biochemical systems. *Phys Biol* 8: 055009.
55. Plimpton S (1995) Fast Parallel Algorithms for Short-Range Molecular Dynamics. *J Comp Phys* 117: 1–19.
56. Sandia National Laboratories (2014) LAMMPS Molecular Dynamics Simulator. Available: <http://lammps.sandia.gov>. Accessed 22 August 2014.
57. Tschernyschkow S, Herda S, Gruenert G, Döring V, Görlich D, et al. (2013) Rule-based modeling and simulations of the inner kinetochore structure. *Prog Biophys Mol Biol* 113: 33–45.
58. MCell (2013) MCell. Available: <http://www.mcell.org>. Accessed 22 August 2014.
59. Kinney JP, Spacek J, Bartol TM, Bajaj CL, Harris KM, et al. (2013) Extracellular sheets and tunnels modulate glutamate diffusion in hippocampal neuropil. *J Comp Neurol* 521: 448–464.
60. Blender Foundation (2014) Blender. Available: <http://www.blender.org>. Accessed 22 August 2014.
61. Shimizu TS, Aksenov SV, Bray D (2003) A spatially extended stochastic model of the bacterial chemotaxis signalling pathway. *J Mol Biol* 329: 291–309.
62. Mirschel S, Steinmetz K, Rempel M, Ginkel M, Gilles ED (2009) PROMOT: modular modeling for systems biology. *Bioinformatics* 25: 687–689.
63. Hoops S, Sahle S, Gauges R, Lee C, Pahle J, et al. (2006) COPASI—a COMplex PATHway Simulator. *Bioinformatics* 22: 3067–3074.
64. Aitken S, Alexander RD, Beggs JD (2013) A rule-based kinetic model of RNA polymerase II C-terminal domain phosphorylation. *J R Soc Interface* 10: 20130438.
65. Artyomov MN, Lis M, Devadas S, Davis MM, Chakraborty AK (2010) CD4 and CD8 binding to MHC molecules primarily acts to enhance Lck delivery. *Proc Natl Acad Sci U S A* 107: 16916–16921.
66. Ibrahim B, Henze R, Gruenert G, Egbert M, Huwald J, et al. (2013) Spatial Rule-Based Modeling: A Method and Its Application to the Human Mitotic Kinetochore. *Cells* 2: 506–544.